

A WSN-based Intelligent Light Control System Considering User Activities and Profiles

Meng-Shiuan Pan, Lun-Wu Yeh, Yen-Ann Chen, Yu-Hsuan Lin, and Yu-Chee Tseng, *Senior Member, IEEE*

Abstract—Recently, wireless sensor networks (WSNs) have been widely discussed in many applications. In this paper, we propose a WSN-based intelligent light control system for indoor environments. Wireless sensors are responsible for measuring current illuminations. Two kinds of lighting devices, namely *whole lighting* and *local lighting* devices, are used to provide background and concentrated illuminations, respectively. Users may have various illumination requirements according to their activities and profiles. An illumination requirement is as the combination of background and concentrated illumination demands and users' locations. We consider two requirement models, namely *binary satisfaction* and *continuous satisfaction* models, and propose two decision algorithms to determine the proper illuminations of devices and to achieve the desired optimization goals. Then a closed-loop device control algorithm is applied to adjust the illumination levels of lighting devices. The prototyping results verify that our ideas are practical and feasible.

Index Terms—intelligent buildings, light control, pervasive computing, wireless communication, wireless sensor network

I. INTRODUCTION

Wireless sensor networks (WSNs) have made a lot of progress recently. Extensive research works have dedicated to energy-efficient MAC protocols [22], sensor deployment and coverage [12], and localization [17]. Applications of WSN include habitat monitoring [3], wildfire monitoring [2], and navigation [13][20].

In this paper, we propose a WSN-based intelligent light control system which considers users' activities and profiles in indoor environments. Fig. 1 shows the network scenario. The network field is divided into regular grids. Each grid has a fixed sensor. Together, these sensors form a multi-hop ad hoc network. One of the nodes is designated as the *sink* of the network and is connected to a *control host*. The control host can issue light control commands via powerline or UPnP communication protocols. In our system, there are two kinds of lighting devices, called *whole lighting* and *local lighting devices*. A whole lighting device is one such as a fluorescent light, which can provide illuminations for multiple grids. For example, in Fig. 1, the light in G_{13} is a whole lighting device,

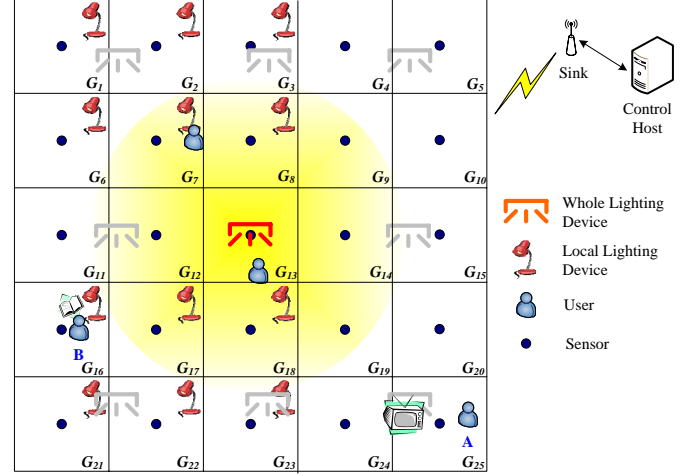


Fig. 1. The network scenario of our system.

which covers grids G_7 , G_8 , G_9 , G_{12} , G_{13} , G_{14} , G_{17} , G_{18} , and G_{19} . A local lighting device is one such as a table lamp, which can only provide concentrated illumination.

In our system, we assume that the location of each user is known and each user carries a wireless sensor, which can detect its local light intensity. Users are considered to have various illumination requirements according to their activities and profiles. For example, in Fig. 1, user *A* is watching television in G_{25} and user *B* is reading in G_{16} . Both *A* and *B* require sufficient background illuminations in their surroundings, and *B* needs concentrated illumination for reading. In this paper, we model an illumination requirement as the combination of background and concentrated lighting according to the user's current activity. An illumination requirement consists of an *illumination interval* and a *coverage range*. A user is said to be *satisfied* if the provided light intensity is in the specified interval for all grids in the coverage range. We further consider a *binary satisfaction* and a *continuous satisfaction* models. In the former, a user who is satisfied returns a satisfaction value of one; otherwise a zero is returned. In the latter model, a satisfaction value which is a function of the specified illumination interval and the sensed light intensity is returned. For the binary model, our goal is to satisfy all users such that the total power consumption is minimized. For the continuous model, our goal is to satisfy all users such that the total satisfaction value is maximized. However, in both models, it may not be possible to satisfy all users simultaneously. In this case, we will gradually relax users' illumination intervals until all users are satisfied. We design

This work is co-sponsored by Taiwan MoE ATU Plan, by NSC grants 93-2752-E-007-001-PAE, 95-2221-E-009-058-MY3, 95-2221-E-009-060-MY3, 96-2219-E-009-007, 96-2218-E-009-004, 96-2622-E-009-004-CC3, and 96-2219-E-007-008, by Realtek Semiconductor Corp., by MOEA under grant number 94-EC-17-A-04-S1-044, by ITRI, Taiwan, by Microsoft Corp., and by Intel Corp.

M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin and Y.-C. Tseng are with the Department of Computer Science, National Chiao-Tung University, Taiwan (e-mail: {mspan, lwyeh, chenya, yuhlin, yctseeng}@cs.nctu.edu.tw)

Y.-C. Tseng is also with the Department of Information and Computer Engineering, Chung-Yuan Christian University, Taiwan

illumination decision algorithms for both models. Then the outputs are sent to a closed-loop device control algorithm to adjust the illuminations of lighting devices. Our prototyping results and system demonstrations verify that our ideas are practical and feasible.

Several works [15][16][19][21] have investigated using WSNs in light control for energy conservation. References [15] and [21] introduce light control using wireless sensors to save energy for commercial buildings. Lighting devices are adjusted according to daylight intensity. Reference [16] defines several kinds of user requirements and their corresponding cost functions. The goal is to adjust lights to minimize the total cost. However, the result is mainly for media production. The work [19] models the light control problem as a trade-off between energy conservation and user requirements. Each user is assigned a utility function with respect to light intensity. The goal is to maximize the total utility. However, it does not consider the fact that people need different illuminations under different activities. Also, some users may suffer from very low utilities, while others enjoy high utilities. In [16][19], it is necessary to measure all combinations of dimmer settings of all devices and the resulting light intensities at all locations. If there are k interested locations, d dimmer levels, and m lighting devices, the complexity is $O(kdm)$. Moreover, the above works only consider one type of lighting devices. In real life, lighting devices can be classified as whole lighting and local lighting ones.

The rest of this paper is organized as follows. Preliminaries are given in Section II. Section III and Section IV introduce our illumination decision algorithms under binary and continuous satisfaction models, respectively. Section V presents our device control algorithm. Section VI reports our prototyping results. Section VII presents some performance evaluation results. Finally, Section VIII concludes this paper.

II. PRELIMINARIES

In this system, there are k grids, n users, m whole lighting devices, and m' local lighting devices. All lighting devices are adjustable. The k grids represent the network area and are labeled as G_1, G_2, \dots , and G_k . In each grid G_i , $i = 1..k$, there is a fixed sensor f_i , and each user u_j , $j = 1..n$, also carries a portable wireless sensor p_j . Users can specify their current activities to the control host via their portable devices. We also assume that via a localization scheme (such as [9]), users' current grid locations are known to the control host.

The whole lighting devices are named D_1, D_2, \dots, D_m , and the local lighting devices are named $d_1, d_2, \dots, d_{m'}$. The fixed sensor that is closest to D_i , $i = 1..m$, is denoted as $f_{c(D_i)}$. However, since users are mobile, we use a function $bound(u_j)$, $j = 1..n$, to denote the association between users and local lighting devices. This function restricts a local lighting device to serve at most one user at one time. If there is no local lighting device near user u_j , $bound(u_j) = \emptyset$; otherwise, $bound(u_j)$ is the ID of the nearest local lighting device. Light intensities sensed by f_i , $i = 1..k$, and p_j , $j = 1..n$, are denoted by $s(f_i)$ and $s(p_j)$, respectively. Since the value of $s(f_i)$ may be contributed by multiple sources, we denote by

$l(D_i)$, $i = 1..m$, the portion of light intensity contributed by D_i to the fixed sensor closest to D_i , i.e., $f_{c(D_i)}$. Note that $l(D_i) \leq s(f_{c(D_i)})$ because $s(f_{c(D_i)})$ may be affected by other whole lighting devices and sunlight. Similarly, we denote by $l(d_i)$, $i = 1..m'$, the portion of light intensity contributed by d_i to portable sensor p_j if user u_j satisfies $bound(u_j) = i$. If there exists no u_j such that $bound(u_j) = i$, we let $l(d_i) = 0$. Note that in reality, the values of $l(D_i)$ and $l(d_i)$ can not be directly known, unless there are no other light sources. We will address this issue in Section II-A.

In the system, sensors periodically report their readings to the sink. For simplicity, we define the following column vectors:

$$\begin{aligned} S_f &= [s(f_1), s(f_2), \dots, s(f_k)]^T, \\ S_p &= [s(p_1), s(p_2), \dots, s(p_n)]^T, \\ L_D &= [l(D_1), l(D_2), \dots, l(D_m)]^T, \\ L_d &= [l(d_1), l(d_2), \dots, l(d_{m'})]^T. \end{aligned}$$

Note that in practice, each D_i has its limitation. So we let $l^{max}(D_i)$ be the upper bound of $l(D_i)$ and let

$$L_D^{max} = [l^{max}(D_1), l^{max}(D_2), \dots, l^{max}(D_m)]^T.$$

We make some assumptions about lighting devices. First, we assume that a local lighting device can always satisfy a user's need when the user is underneath this device. Second, we assume that there is no obstacle between whole lighting devices and fixed sensors. Third, the illumination provided by a local lighting device does not affect the measured light intensity of fixed sensors.

Fig. 2 shows our system architecture. Light adjustments are triggered by users' movements or environment changes. First, the illuminations of whole lighting devices are determined, followed by those of the local lighting devices. Feedbacks from sensors are then sent to the sink to decide further adjustment of lighting devices so as to satisfy users' demands.

A. Computing L_D and L_d

Earlier, we mentioned that the values of L_D and L_d can not be known directly. Below, we first use an experimental method to derive L_D . Assuming no other light source existing, Fig. 3(a) shows the measured intensities of a whole lighting device D_i by $f_{c(D_i)}$ and other fixed sensors at different distances from $f_{c(D_i)}$, under different on-levels of D_i . We see that the measured intensity degrades following a similar trend. In fact, if we further normalize the value to the intensity measured by $f_{c(D_i)}$, we see that the degrading trends are almost the same, as shown in Fig. 3(b). Therefore, assuming the impact factor of D_i on $f_{c(D_i)}$ to be $w_{c(D_i)}^i = 1$, the impact factor of D_i on any other f_j can be written as a weighted factor w_j^i , where $0 \leq w_j^i \leq 1$. Putting all impact factors together, we define a weight matrix

$$W = \begin{bmatrix} w_1^1 & w_1^2 & \dots & w_1^m \\ w_2^1 & w_2^2 & \dots & w_2^m \\ \vdots & \vdots & \ddots & \vdots \\ w_k^1 & w_k^2 & \dots & w_k^m \end{bmatrix}.$$

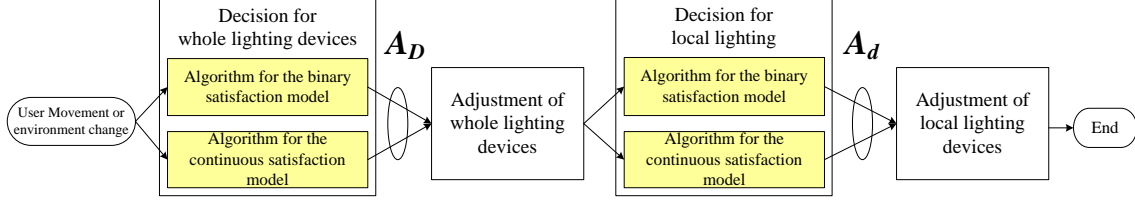


Fig. 2. The system architecture of our light control system.

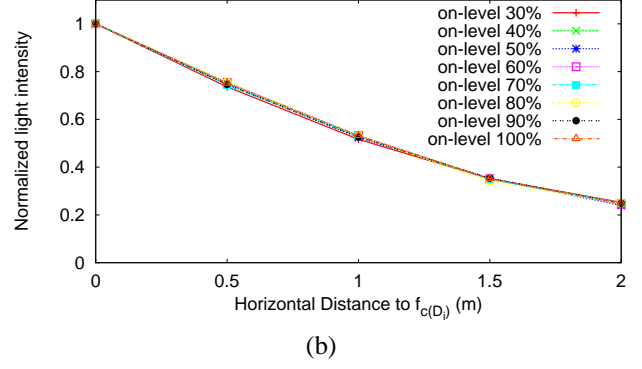
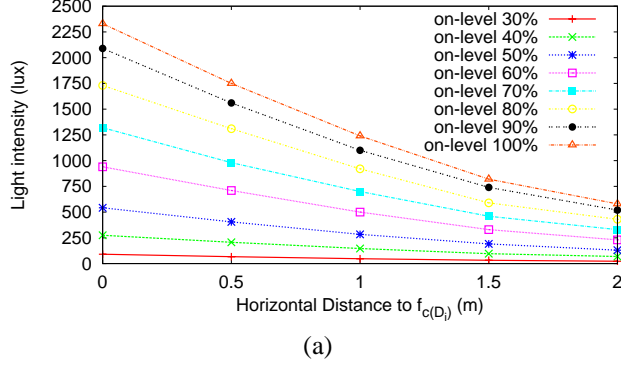


Fig. 3. An experiment for characterizing the degradation of light signals.

Since light intensities are additive [19], the light intensity measured by $f_{c(D_i)}$ is the sum of intensities from sunlight, D_i , and neighboring devices. The intensities of the sunlight to all fixed sensors are written as a $k \times 1$ column vector S_{sun} . So we have

$$S_f = W \cdot L_D + S_{sun}. \quad (1)$$

In Eq. (1), there are m unknowns in L_D and k equations, where $m \leq k$. Any typical k -means algorithm [14] can solve Eq. (1) by inducing the least mean square error. Here, we simply construct a new $m \times m$ matrix \hat{W} by keeping all $c(D_i)$ -th rows, $i = 1..m$, in W and removing the other $k - m$ rows. So, Eq. (1) can be rewritten as

$$S_f - S_{sun} = \hat{W} \cdot L_D \Rightarrow L_D = \hat{W}^{-1} \cdot (S_f - S_{sun}). \quad (2)$$

The weight matrix W can be measured at the deployment stage, vector S_{sun} can be measured on-line when all lights are off, and vector S_f can be obtained on-line. So the calibration complexity is $O(km)$. This is lower than those of [16][19].

The calculation of L_d is quite straightforward. Due to the property of our approach, before a user arrives at a d_i , no measurement can be obtained for $l(d_i)$. At this time, $l(d_i) = 0$. When a portable sensor, say, p_k is getting close to and bounded with d_i , the local lighting device d_i may be triggered. Here, we simply use the reading of the fixed sensor, say, f_j located at the same grid as d_i as the background light intensity. We let the light intensity provided by d_i to p_k be

$$l(d_i) = s(p_k) - s(f_j).$$

III. SOLUTION FOR THE BINARY SATISFACTION MODEL

Each user profile consists of a number of activity-requirement pairs. Given an activity, the system should try

to satisfy the corresponding requirement. Each requirement of a user u_i has three parts:

- 1) Expected illumination interval of whole lighting: $[B_D^l(u_i), B_D^u(u_i)]$ (in lux), where $B_D^l(u_i)$ and $B_D^u(u_i)$ are the lower and the upper bounds, respectively.
- 2) Expected illumination interval of local lighting: $[B_d^l(u_i), B_d^u(u_i)]$, where $B_d^l(u_i)$ and $B_d^u(u_i)$ are the lower and the upper bounds, respectively.
- 3) Coverage range of whole lighting: $R_i = [r_i(G_1), r_i(G_2), \dots, r_i(G_k)]^T$, where for each $j = 1..k$, $r_i(G_j) = 1$ if grid G_j is expected to receive a light intensity within $[B_D^l(u_i), B_D^u(u_i)]$ for user u_i ; otherwise, $r_i(G_j) = 0$. This array defines the range of grids which should meet the whole lighting requirement.

For example, a possible requirement of a reading user B in Fig. 1 can be $[B_D^l(u_B), B_D^u(u_B)] = [200, 600]$, $[B_d^l(u_B), B_d^u(u_B)] = [500, 1000]$, and $R_B = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0]^T$.

Let L_D and L_d be the current intensity vectors provided by whole and local lighting devices, respectively. To facilitate the presentation, let $X_K = [1 \ 1 \ \dots \ 1]$ be a $1 \times K$ row vector, and \bar{R}_i a $k \times k$ matrix such that

$$\bar{R}_i = \begin{bmatrix} r_i(G_1) & 0 & \dots & 0 \\ 0 & r_i(G_2) & \dots & 0 \\ 0 & \vdots & \dots & \vdots \\ 0 & 0 & \dots & r_i(G_k) \end{bmatrix}.$$

We formulate our problem P as a linear programming problem with inputs S_f , S_p , L_D , L_d , W , and user requirements. Our

goal is to find the adjustment vectors

$$A_D = [a(D_1), a(D_2), \dots, a(D_m)]^T$$

$$A_d = [a(d_1), a(d_2), \dots, a(d_{m'})]^T$$

for whole and local lighting devices, respectively, where $a(D_i)$, $i = 1..m$, and $a(d_j)$, $j = 1..m'$, are the amounts of adjustment required for D_i and d_j , respectively, such that the following two objectives are satisfied:

$$\min X_m(A_D + L_D) \quad (3)$$

$$\min X_{m'}(A_d + L_d) \quad (4)$$

subject to:

$$B_D^l(u_i)R_i \leq \bar{R}_i(S_f + WA_D) \leq B_D^u(u_i)R_i, \forall i \in [1, n] \quad (5)$$

$$O \leq A_D + L_D \leq L_D^{max} \quad (6)$$

$$B_d^l(u_i) \leq a(d_j) + s(p_i) \leq B_d^u(u_i),$$

$$\text{if } bound(u_i) = j, \forall i \in [1, n]. \quad (7)$$

Eq. (3) and Eq. (4) mean that the total power consumptions of both whole and local lighting devices after the adjustment should be minimized. Eq. (5) imposes the whole lighting requirement, where $S_f + WA_D$ is the light intensity vector after adjustment and matrix \bar{R}_i is to filter out those grids not in the coverage range of whole lighting. Eq. (6) is to confine the adjustment result within the maximum and the minimum capacities of devices, where O is a zero vector. Eq. (7) is to impose the requirement of each local lighting if a user is bounded to it. Here we assume that local lighting can always provide extra illuminations to satisfy users' requirements. So we do not specify upper bounds as that in Eq. (6).

Since we assume that the illuminations of local lighting devices do not affect the measured light intensity of fixed sensors, the decision of whole lighting levels can be made independently of the decision of local lighting levels. (However, the reverse is not true because the decision of whole lighting levels does affect the decision of local lighting levels.) This allows us to solve problem P in two stages as formulated below.

P1: Given S_f , L_D , W , and user requirements, solve A_D for Eq. (3), Eq. (5), and Eq. (6).

P2: Given S_p , L_d , and user requirements, solve A_d for Eq. (4) and Eq. (7).

Theorem 1: Problem P is equivalent to the joint problems $P1$ and $P2$.

Problem $P1$ is a linear programming problem, which can be solved by the Simplex method [11], unless the problem itself is infeasible, which may happen when two users have conflicting requirements on the same grid. When no feasible solution can be found, our system will try to eliminate some constraints to make $P1$ feasible. Reference [18] already shows that finding a feasible subsystem of a linear system by eliminating the fewest constraints is NP-hard. Hence, we propose a heuristic below.

The idea is to gradually relax some requirements until a feasible solution appears. We first define some notations. Given the current values of S_f , L_D , and L_D^{max} , it is easy to compute the minimum and maximum possible illuminations of grids

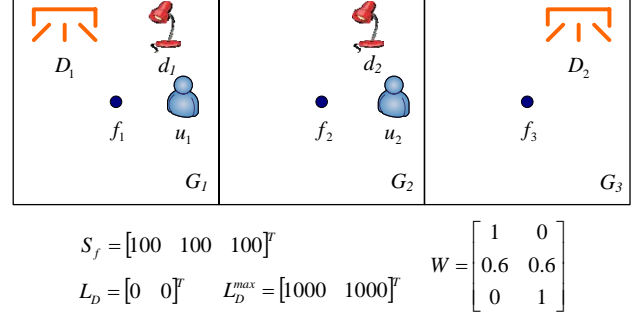


Fig. 4. An example of illumination decision.

by $S_f^{min} = S_f - WL_D$ and $S_f^{max} = S_f + W(L_D^{max} - L_D)$. Also, consider c intervals on \mathbb{R} (the set of reals) which define c users' requirements on whole lighting. We say that an interval $[a, b] \in \mathbb{R}$ has an *overlapping degree* of d if for each point $p \in [a, b]$, p falls in at least d of the above c intervals. An interval $[a, b]$ is said to be a *max-interval* if there exists no other interval $[a', b']$ which has a higher overlapping degree than $[a, b]$ and $[a', b']$ is a superset of $[a, b]$. It is not hard to see that given any c intervals, there must exist a max-interval. Also it is easy to design a polynomial-time linear search algorithm to find a max-interval (we omit the details here). Our algorithm works as follows.

- 1) For each grid G_i , $i = 1..k$, find the set of users U_i whose coverage ranges contain G_i , i.e., $U_i = \{u_j | r_j(G_i) = 1, \forall j \in [1, n]\}$. For each user $u_j \in U_i$, check if $[B_D^l(u_j), B_D^u(u_j)] \cap [S_f^{min}[i], S_f^{max}[i]] = \emptyset$. If so, the requirement cannot be satisfied. So we set $r_j(G_i) = 0$ and update \bar{R}_j .
- 2) Again, for each grid G_i , $i = 1..k$, consider the set U_i . Check if there is a common overlapping interval for the requirements of all users in U_i . If not, find a max-interval, say, $[a, b]$ for the requirements of all users in U_i . For each user $u_j \in U_i$, check if $[B_D^l(u_j), B_D^u(u_j)] \cap [a, b] = \emptyset$. If so, we will give up the requirement of u_j . So we set $r_j(G_i) = 0$ and update \bar{R}_j .
- 3) Try to solve problem $P1$. If there exists no feasible solution A_D , relax the whole lighting requirement of each user u_i , $i = 1..n$, to $[B_D^l(u_i) - \alpha, B_D^u(u_i) + \alpha]$, where α is a predefined constant. Then repeat this step again.
- 4) After deciding A_D , solve problem $P2$ as follows. For each d_j , $j = 1..m'$, check if there is a user u_i such that $bound(u_i) = j$. If so, set $a(d_j) = B_d^l(u_i) - s(p_i)$; otherwise, we can inform the system to turn d_j off.

Example 1: Fig. 4 shows a scenario with three grids, two users, two whole lighting devices, and two local lighting devices. User u_1 's requirements are $[B_D^l(u_1), B_D^u(u_1)] = [200, 400]$, $[B_d^l(u_1), B_d^u(u_1)] = [700, 900]$, and $R_1 = [1, 0, 0]^T$. User u_2 's requirements are $[B_D^l(u_2), B_D^u(u_2)] = [300, 500]$, $[B_d^l(u_2), B_d^u(u_2)] = [800, 1000]$, and $R_2 = [0, 1, 0]^T$. Problem $P1$ has the objective:

$$\begin{aligned} \min \quad & [1 \quad 1] \left(\begin{bmatrix} a(D_1) \\ a(D_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) \\ \equiv \quad & \min (a(D_1) + a(D_2)) \end{aligned}$$

subject to:

$$200 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0.6 & 0.6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a(D_1) \\ a(D_2) \end{bmatrix} \right) \leq 400 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\equiv \begin{bmatrix} 200 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 100 + a(D_1) \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 400 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 300 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 100 + 0.6a(D_1) + 0.6a(D_2) \\ 0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 500 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} a(D_1) \\ a(D_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1000 \\ 1000 \end{bmatrix}.$$

Since $P1$ is feasible, the results are $a(D_1) = 184$ and $a(D_2) = 150$.

After adjusting whole lighting devices, $S_p = [s(p_1), s(p_2)]^T = [284, 300]^T$ and $L_d = [l(d_1), l(d_2)]^T = [0, 0]^T$. So problem $P2$ has the objective:

$$\begin{aligned} \min \quad & [1 \quad 1] \left(\begin{bmatrix} a(d_1) \\ a(d_2) \end{bmatrix} + \begin{bmatrix} 284 \\ 300 \end{bmatrix} \right) \\ \equiv \quad & \min (a(d_1) + a(d_2) + 584) \end{aligned}$$

subject to:

$$\begin{aligned} 700 &\leq a(d_1) + 284 \leq 900 \\ 800 &\leq a(d_2) + 300 \leq 1000. \end{aligned}$$

The adjustments of local lighting devices are as $a(d_1) = B_d^l(u_1) - s(p_1) = 416$ and $a(d_2) = B_d^l(u_2) - s(p_2) = 500$.

IV. SOLUTION FOR THE CONTINUOUS SATISFACTION MODEL

In this model, a user's requirement on illumination is not a fixed interval. Instead, it is a continuous value where each value is mapped to a satisfaction level. User u_i 's requirement has four parts:

- 1) Satisfaction level of whole lighting, which is represented by a modified Gaussian distribution by normalizing the peak value to 1 with mean and variance $(\mu_D(u_i), \sigma_D(u_i))$. Specifically, the satisfaction level of intensity x is $c_D(u_i, x) = \exp(\frac{-(x - \mu_D(u_i))^2}{2(\sigma_D(u_i))^2})$.
- 2) Satisfaction threshold of whole lighting: \bar{t} , $0 \leq \bar{t} < 1$. That is, after the adjustment, the satisfaction level must be no less than \bar{t} . From \bar{t} , we can derive the desired illumination interval of whole lighting $[B_D^l(u_i, \bar{t}), B_D^u(u_i, \bar{t})] = [\mu_D(u_i) - \sigma_D(u_i)\sqrt{-2\ln(\bar{t})}, \mu_D(u_i) + \sigma_D(u_i)\sqrt{-2\ln(\bar{t})}]$.

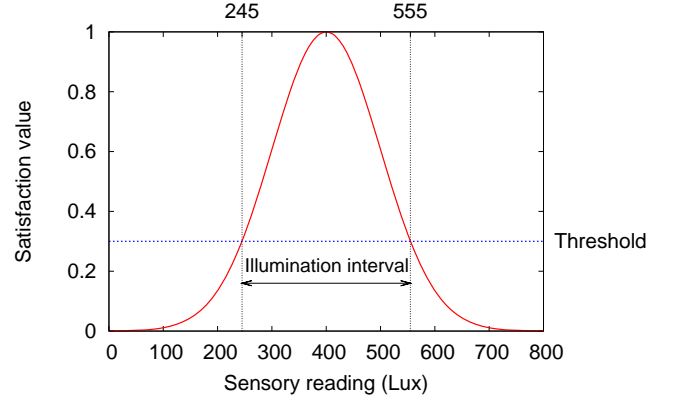


Fig. 5. Example of user's satisfaction level, where $(\mu_D(u_i), \sigma_D(u_i)) = (400, 100)$.

- 3) Satisfaction level of local lighting, which is represented by a modified Gaussian distribution by normalizing the peak value to 1 with mean and variance $(\mu_d(u_i), \sigma_d(u_i))$. Specifically, the satisfaction level of intensity x is $c_d(u_i, x) = \exp(\frac{-(x - \mu_d(u_i))^2}{2(\sigma_d(u_i))^2})$.
- 4) coverage range of whole lighting: \bar{R}_i .

For example, Fig. 5 shows a satisfaction level. Given $\bar{t} = 0.3$, $[B_D^l(u_i), B_D^u(u_i)] = [245, 555]$.

Our goal is to find the adjustment vectors A_D and A_d such that the total satisfaction level of all users is maximized. Here, the satisfaction level of a user u_i is sum of the satisfaction level of u_i at each grid G_j , $j = 1..k$, such that $r(G_j) = 1$. Recall that $S_f + W A_D$ is the intensities perceived by all fixed sensors. We define $C'(u_i, S_f + W A_D)$ as the vector of satisfaction levels of u_i at all grids, i.e.,

$$C'(u_i, S_f + W A_D)[j] = c_D(u_i, (S_f + W A_D)[j]).$$

Therefore, given S_f , S_p , L_D , L_d , and user requirements, we can formulate a nonlinear programming problem with objectives:

$$\max \sum_{i=1}^n (R_i)^T \cdot C'(u_i, S_f + W A_D) \quad (8)$$

$$c_d(u_i, a(d_j) + s(p_i)) = 1 \text{ if } \text{bound}(u_i) = j, \forall i \in [1, n] \quad (9)$$

subject to:

$$B_D^l(u_i, \bar{t}) R_i \leq \bar{R}_i (S_f + W A_D) \leq B_D^u(u_i, \bar{t}) R_i, \forall i \in [1, n] \quad (10)$$

$$0 \leq A_D + L_D \leq L_D^{max} \quad (11)$$

Eq. (8) is to maximize the sum of satisfaction levels of all users. Eq. (9) is so written because we assume that local lighting devices can always maximize users' local lighting satisfaction levels. Eq. (10) and Eq. (11) are the same as the ones in Section III.

Again, the above nonlinear programming problem can be solved in two stages:

P3: Given S_f , L_D , W , and user requirements, solve A_D for Eq. (8), Eq. (10), and Eq. (11).

P4: Given S_p , L_d , and user requirements, solve A_d for Eq. (9). $P3$ can be solved by a sequential quadratic programming

(SQP) method [10]. The basic idea is as follows. It first reformulates the problem into a quadratic programming subproblem using an approximate solution x^k . Then, it uses x^k to construct a better approximation x^{k+1} . The process will eventually converge to an optimal solution x^* , unless $P3$ is infeasible. If so, we will gradually decrease the threshold \bar{t} until there is a solution can be found. Given the S_f^{min} and S_f^{max} as defined in Section III, the detail algorithm works as follows.

- 1) For each grid $G_i, i = 1..k$, find the set of users U_i whose coverage ranges contain G_i , i.e., $U_i = \{u_j | r_j(G_i) = 1, \forall j \in [1, n]\}$. For each user $u_j \in U_i$, check if $[B_D^l(u_j, \bar{t}), B_D^u(u_j, \bar{t})] \cap [S_f^{min}[i], S_f^{max}[i]] = \emptyset$. If so, the requirement cannot be satisfied. So we set $r_j(G_i) = 0$ and update \bar{R}_j .
- 2) Again, for each grid $G_i, i = 1..k$, consider the set U_i . Check if there is a common overlapping interval for the requirements of all users in U_i . If not, find a max-interval, say, $[a, b]$ for the desired illumination intervals of all users in U_i . For each user $u_j \in U_i$, check if $[B_D^l(u_j, \bar{t}), B_D^u(u_j, \bar{t})] \cap [a, b] = \emptyset$. If so, we will give up the requirement of u_j . So we set $r_j(G_i) = 0$ and update \bar{R}_j .
- 3) Try to solve problem $P3$ by SQP. If there exists no feasible solution A_D , relax the \bar{t} to $\bar{t} - \gamma$, where γ is a predefined constant. Then repeat this step again.
- 4) After deciding A_D , solve problem $P4$ as follows. For each local $d_j, j = 1..m'$, check if there is a user u_i such that $bound(u_i) = j$. If so, find a value of $a(d_j)$ such that $p_d(u_i, a(d_j) + s(p_i)) = 1$; otherwise, we can inform the system to turn d_j off.

Example 2: Let's use Fig. 4 again by assuming $(\mu_D(u_1), \sigma_D(u_1)) = (300, 100)$, $(\mu_d(u_1), \sigma_d(u_1)) = (800, 100)$, $R_1 = [1, 0, 0]^T$, $(\mu_D(u_2), \sigma_D(u_2)) = (400, 100)$, $(\mu_d(u_2), \sigma_d(u_2)) = (1000, 100)$, and $R_2 = [0, 1, 0]^T$. Given $\bar{t} = 0.3$, we can have $[B_D^l(u_1, \bar{t}), B_D^u(u_1, \bar{t})] = [145, 455]$ and $[B_D^l(u_2, \bar{t}), B_D^u(u_2, \bar{t})] = [300, 500]$. Problem $P3$ has the objective.

$$\begin{aligned}
& \max [1 \ 0 \ 0] C'_1 \left(u_1, \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0.6 & 0.6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a(D_1) \\ a(D_2) \end{bmatrix} \right) \\
& + [0 \ 1 \ 0] C'_2 \left(u_2, \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0.6 & 0.6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a(D_1) \\ a(D_2) \end{bmatrix} \right) \\
& \equiv \max [1 \ 0 \ 0] C'_1 \left(u_1, \begin{bmatrix} 100 + a(D_1) \\ 100 + 0.6a(D_1) + 0.6a(D_2) \\ 100 + a(D_2) \end{bmatrix} \right) \\
& + [0 \ 1 \ 0] C'_2 \left(u_2, \begin{bmatrix} 100 + a(D_1) \\ 100 + 0.6a(D_1) + 0.6a(D_2) \\ 100 + a(D_2) \end{bmatrix} \right) \\
& \equiv \max c_D(u_1, 100 + a(D_1)) \\
& \quad + c_D(u_2, 100 + 0.6a(D_1) + 0.6a(D_2))
\end{aligned}$$

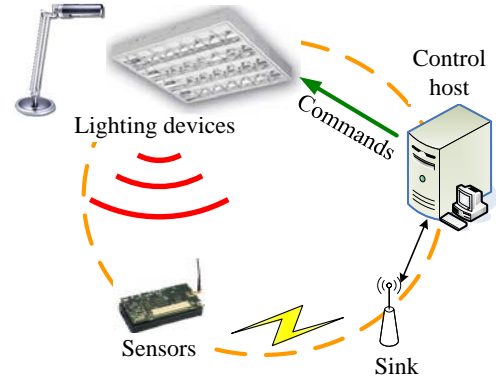


Fig. 6. The closed-loop device control procedure.

subject to:

$$\begin{aligned}
& (300 - 100\sqrt{-2 \ln 0.3}) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \leq \\
& \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0.6 & 0.6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a(D_1) \\ a(D_2) \end{bmatrix} \right) \\
& \leq (300 + 100\sqrt{-2 \ln 0.3}) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
& \equiv \begin{bmatrix} 145 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 100 + a(D_1) \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 455 \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} 0 \\ 245 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 100 + 0.6a(D_1) + 0.6a(D_2) \\ 0 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 555 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} a(D_1) \\ a(D_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1000 \\ 1000 \end{bmatrix}
\end{aligned}$$

After applying SQP, the result is $a(D_1) = 200$ and $a(D_2) = 300$.

After adjusting whole lighting device, $S_p = [s(p_1), s(p_2)]^T = [300, 400]^T$. To let $c_d(u_1, a(d_1) + s(p_1)) = c_d(u_1, a(d_1) + 300) = 1$, the adjustment of d_1 is as $a(d_1) = 500$. Similarly, $a(d_2) = 600$.

V. DEVICE CONTROL ALGORITHM

Given the light intensities contributed by devices to sensors, i.e., L_D and L_d , the algorithms in Section III and Section IV will determine the target adjustment amounts, i.e., A_D and A_d . However, since what reported by sensors are accumulated values, we have to convert these values to the actual adjustment amounts. If the actual amounts do not match the target amounts, we will adopt a binary search technique to gradually approach these amounts.

Below, let $L_D^{(1)}$ and $L_d^{(1)}$ be the current contributed intensities of whole and local lighting devices, respectively, and $L_D^{(*)} = L_D^{(1)} + A_D$ and $L_d^{(*)} = L_d^{(1)} + A_d$ be the target ones.

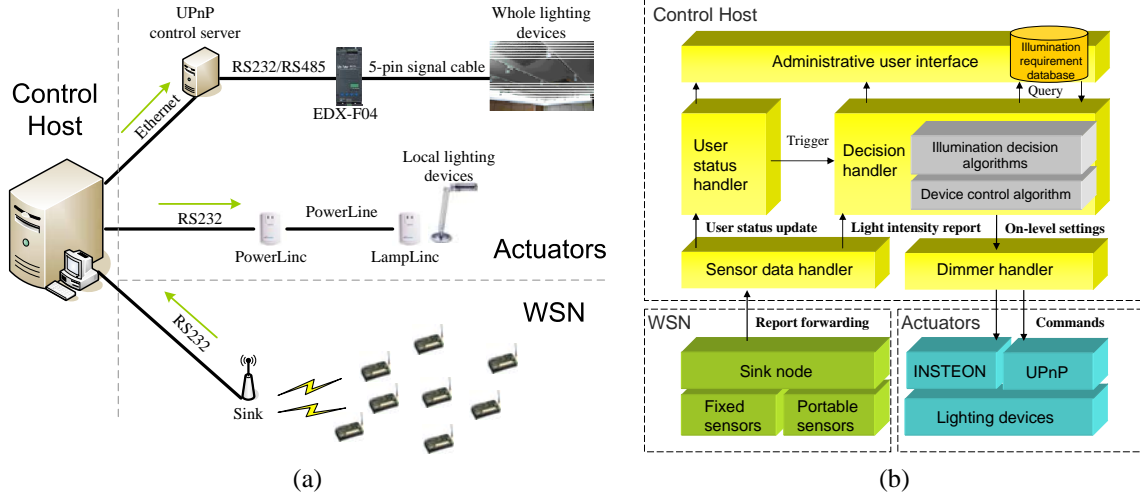


Fig. 7. (a) System architecture and (b) components of our intelligent light control system.

Our algorithm contains multiple iterations. In the i -th iteration, $i \geq 1$, based on $L_D^{(i)}$ and $L_d^{(i)}$, we will adjust devices leading to new intensities $L_D^{(i+1)}$ and $L_d^{(i+1)}$. This will be repeated until the target values are reached or no further improvement is possible. Such a closed loop control is illustrated in Fig. 6. The binary search procedure can be explained by the following example. Suppose that device D_i 's current on-level is 40% with contribution $l^{(0)}(D_i) = 300$ lux to sensor $f_{c(D_i)}$ and $l^{(*)}(D_i) = 200$ lux. The control host will first adjust the on-level of D_i to $(0 + 40)/2 = 20\%$. After first iteration, the control host will collect sensors' reports to compute $L_D^{(1)}$ and thus $l^{(1)}(D_i)$. With $l^{(1)}(D_i)$, the next guess will be an on-level of 10% or 30%. The similar trial will be done for all whole and local lighting devices.

In practice, the on-levels of dimmers are discrete and have finite levels. The termination conditions of the above binary search can be controlled by a threshold, say, β when $|l^{(i+1)}(D_j) - l^{(i)}(D_j)| \leq \beta$. To accelerate the decision, the control host can even record the relationship between the contributed light intensities and on-levels of devices (we omit the details here).

VI. PROTOTYPING RESULTS

This section presents our implementation of the intelligent light control system. Fig. 7 shows the system architecture and the related protocol components. The system can be divided into three parts: wireless sensor network, actuators, and control host. In the following, we describe each part in details.

A. Wireless Sensor Network

Our sensor nodes are developed using Jennic JN5121 [4] as the radio module and Si photodiode IC [6] as the photo sensor (Fig. 8). Users can indicate their current activities to the system by clicking the buttons on the sensor board. Fixed sensors are used to form the backbone of the network. A portable sensor will associate with the nearest fixed sensor. Fixed and portable sensors periodically report aggregated light intensity values to the sink. The sink forwards sensing data to the control

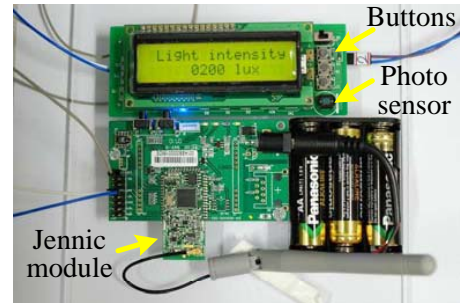


Fig. 8. The implemented sensor board.

host via an RS232 interface. Note that when a sensor finds that its surrounding light intensity changes rapidly, it will also report. This happens when the control host is adjusting lighting devices. Moreover, we implement a reduced version of the localization scheme in [9] to trace users' locations. Once a portable sensor decides its owner's location, it issues a location update to the control host.

B. Actuators

In our current implementation, whole and local lighting devices are controlled by different ways. We implement the UPNP Lighting Controls V1.0 standard [8] to control whole lighting devices. The control host issues UPNP device control commands to the UPNP control server through the Internet. Then the UPNP control server controls some dimmer EDX-F04 dimmers [1], which are connected to whole lighting devices. On the other hand, we use the INSTEON LampLinc dimmer and PowerLinc controller manufactured by SmartHome [7] to control local lighting devices. Each local lighting device is plugged in a LampLinc dimmer. The PowerLinc controller is connected to the control host. When receiving control commands from the control host, the PowerLinc controller can control dimmers through the power-line network.

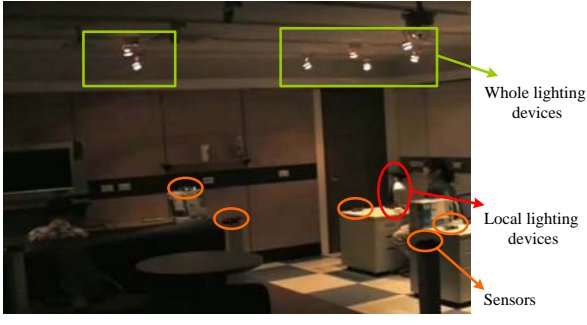


Fig. 9. The demonstration scenario of our intelligent light control system.

C. Control Host

The control host is implemented by Java. It consists of five components.

- *Sensor data handler*: Its main task is to classify the report data from the sink into two types: *user status update* and *light intensity report*. Then it relays these data to the corresponding components.
- *User status handler*: This component tracks the latest locations and activities of users. When detecting any change of users' locations or activities, it triggers the decision handler component to compute new illumination requirements.
- *Decision handler*: This component implements the algorithms in Section III, Section IV, and Section V. It is triggered by the user status handler component or by any change in the environment. We use Matlab to implement our algorithms in Section III and Section IV. The Matlab program is translated to a Java program by the Matlab builder for Java [5]. After making device control decisions, it sends on-level settings to the dimmer handler.
- *Dimmer handler*: This component serves as the interface between the control host and the actuators and issues commands to the UPnP control server and the INSTEON PowerLinc controller.
- *Administrative user interface*: We implement a graphical user interface (GUI), which contains three panels: 1) The *monitor panel* shows the locations of users, fixed sensors, and lighting devices. 2) The *configuration panel* is for the system manager to plan the network and set system parameters. 3) The *information panel* shows the reported sensory readings, the connection statuses of sensor nodes, and so on.

Fig. 9 shows the demo scenario of our system. We build the light control system in a room of size $5\text{ m} \times 5\text{ m}$, which is divided into 3×3 grids. More details and demo videos can be found in <http://wsn-research.blogspot.com/>.

VII. PERFORMANCE EVALUATIONS

We use some experiments and simulations to verify our results.

A) *Verification of the estimation of L_D* : In Section II-A, we show how to evaluate L_D . Here we use the network scenario in Fig. 10 with 12 grids and three whole lighting devices to

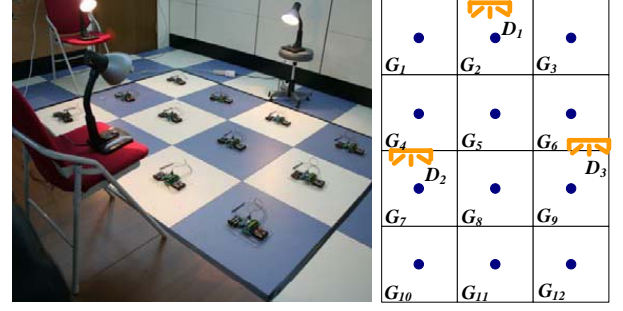


Fig. 10. The scenario to verify the measured L_D .

verify the result. Here, we simply use lamps as whole lighting devices. With different on-levels for lamps, we compute L_D and compare it against the actual measured value. Fig. 11 shows the comparison without and with sunlight effect. We can see that the computed and the measured values are quite close.

B) *Verification of the binary satisfaction model (BSM)*: We set up two scenarios, $S1$ and $S2$. Scenario $S1$ has 5×5 grids with 9 whole lighting devices as in Fig. 1. Scenario $S2$ has 9×9 grids with 25 whole lighting devices. In both scenarios, each whole lighting device can cover its nearby 9 grids. The weighted factors of each whole lighting device D_i on nearby fixed sensors are set as follows. (1) The weighted factor of D_i on the fixed sensor at $G_{c(D_i)}$ is 1. (2) For fixed sensors in left, right, up, and down grids of $G_{c(D_i)}$, the weights are set to 0.5. (3) For fixed sensors in upper-left, lower-left, upper-right, and lower-right grids of $G_{c(D_i)}$, the weights are 0.25. (4) For all other fixed sensors, the weights are 0. Local lighting devices are not simulated since they have no impact on performance. All lighting devices are initially set to be turned off.

We define two activity-requirement pools, called $AR1$ and $AR2$, as shown in Fig. 12. Each act_i in Fig. 12 represents an expected illumination interval of whole lighting. In our simulations, users randomly select their activities from a pool. The coverage range of a user's requirement is the five nearest grids. We compare our algorithm against a fixed adjustment scheme (denoted by FIX), where lighting devices are set to fixed levels. If a user's requirement coverage range overlaps a lighting device's coverage range, this device is turned to that level. Below, we use FIX- n to indicate that each device can provide at most n lux.

We consider two performance indices. First, considering that our algorithm may enlarge users' illumination requirements when conflicts occur, we define a metric GAP to represent the difference between the provided light intensity and the original requirement of a user. For user u_i with coverage range R_i , if grid G_j satisfies $r_i(G_j) = 1$, we compute a gap value as

$$gap(u_i, G_j) = \begin{cases} 0 & \text{if } B_D^l(u_i) \leq s(f_j) \leq B_D^u(u_i) \\ \min(|B_D^l(u_i) - s(f_j)|, |B_D^u(u_i) - s(f_j)|) & \text{o.w.,} \end{cases}$$

where $s(f_j)$ is the final sensory value of f_j . Then we define GAP of u_i as the average of $gap(u_i, G_j)$ for all G_j such that $r_i(G_j) = 1$. The second index is $X_m A_D$, which represents the energy consumption of one control decision.

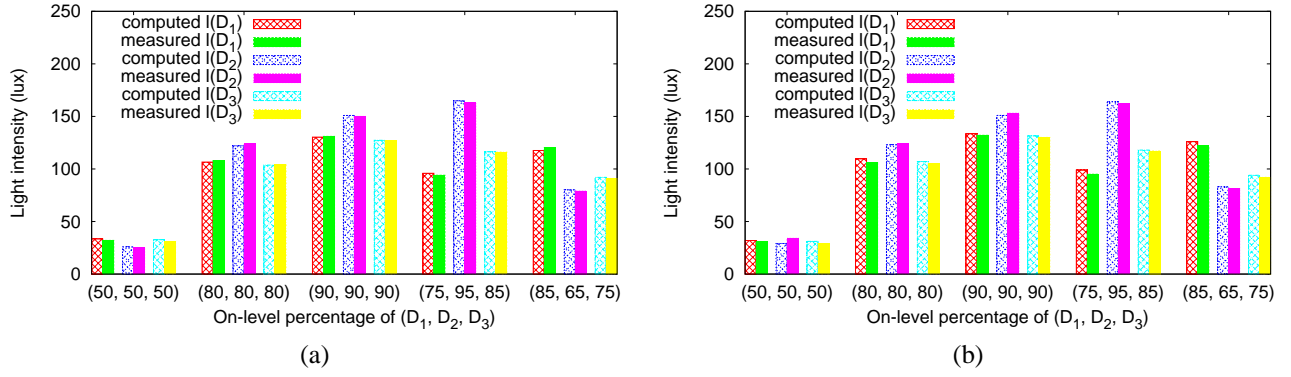


Fig. 11. Experiments on computed and measured L_D when the environment is (a) without and (b) with sunlight effect.

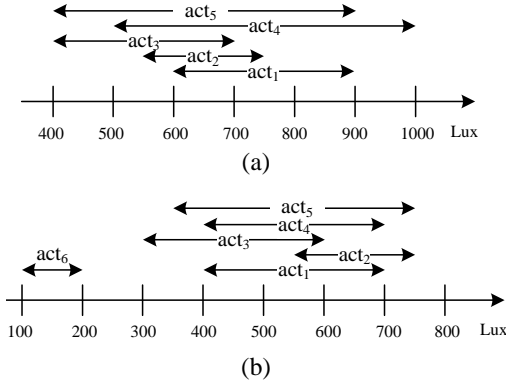


Fig. 12. Activity-requirement pools: (a) $AR1$ and (b) $AR2$.

Fig. 13(a), Fig. 13(b), Fig. 13(c), and Fig. 13(d) show our simulation results under different combinations of $S1/S2$ and $AR1/AR2$. In the left figure of Fig. 13(a), we see that the average GAP of users is almost zero for BSM. This is because the illumination intervals in $AR1$ have common overlapping, which allows our algorithm to satisfy all users in most cases. The right figure of Fig. 13(a) compares the energy consumption of different schemes. FIX-500 has a slightly lower value than ours because some users' requirements are violated. Fig. 13(b) adopts $AR2$. Since some requirements are violated, we see that our scheme also induces some gaps (note that act_6 has no overlapping with others). In terms of energy cost, BSM outperforms the other schemes. Fig. 13(c) and Fig. 13(d) adopt $S2$ and the trends are similar. This demonstrates that our scheme is quite scalable to network size.

C) Verification of the continuous satisfaction model (CSM): We also define two activity-requirement pools, called $AR3$ and $AR4$, as shown in Fig. 14. The satisfaction threshold \bar{t} of whole lighting is set to 0.3. Similarly, users' required coverage range of whole lighting is the five nearest grids. We compare two performance indices: users' average satisfaction level and energy consumption.

Fig. 15(a), Fig. 15(b), Fig. 15(c), and Fig. 15(d) show our simulation results under different combinations of $S1/S2$ and $AR3/AR4$. These results consistently indicates that our scheme provides the highest satisfaction levels and outperforms FIX-750 and FIX-1000 in energy cost. Note that Fix-

500 may save some energy at the cost of users' satisfaction. Also note that $AR4$ has higher deviation in requirements than $AR3$.

VIII. CONCLUSIONS

In this paper, we have presented a WSN-based intelligent light control system considering user activities and profiles. In this system, there are two types of lighting devices. We use wireless sensors to collect light intensities in the environment. Considering users' activities, we model the illumination requirements of users. Illumination decision algorithms and a device control algorithm are presented to meet user requirements and to conserve energy. The proposed schemes are verified by real implementation in an indoor environment. Future directions could be directed to relieving the computation cost of the non-linear programming and enhance the user interfaces at the portable sensor nodes.

REFERENCES

- [1] Edx-f04. <http://www.liteputer.com.cn/china/liteputer-tw/product.asp>.
- [2] Design and construction of a wildfire instrumentation system using networked sensors. <http://firebug.sourceforge.net/>.
- [3] Habitat monitoring on great duck island. <http://www.greatduckisland.net/technology.php>.
- [4] Jennic JN5121. <http://www.jennic.com/>.
- [5] Matlab Builder for Java. <http://www.mathworks.com/products/javabuilder/>.
- [6] Si photodiode s1133. <http://jp.hamamatsu.com/en/index.html>.
- [7] SmartHome Inc. <http://www.smarthome.com>.
- [8] UPnP forum. <http://www.upnp.org>.
- [9] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. of IEEE INFOCOM*, 2000.
- [10] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 45(1):1–51, 1995.
- [11] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 2001.
- [12] C.-F. Huang, Y.-C. Tseng, and L.-C. Lo. The coverage problem in three-dimensional wireless sensor networks. *Journal of Interconnection Networks*, 8(3):209–227, 2007.
- [13] Q. Li, M. DeRosa, and D. Rus. Distributed algorithm for guiding navigation across a sensor network. In *Proc. of ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Maryland, USA, 2003.
- [14] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [15] F. O'Reilly and J. Buckley. Use of wireless sensor networks for fluorescent lighting control with daylight substitution. In *Proc. of Workshop on Real-World Wireless Sensor Networks (REANWSN)*, 2005.

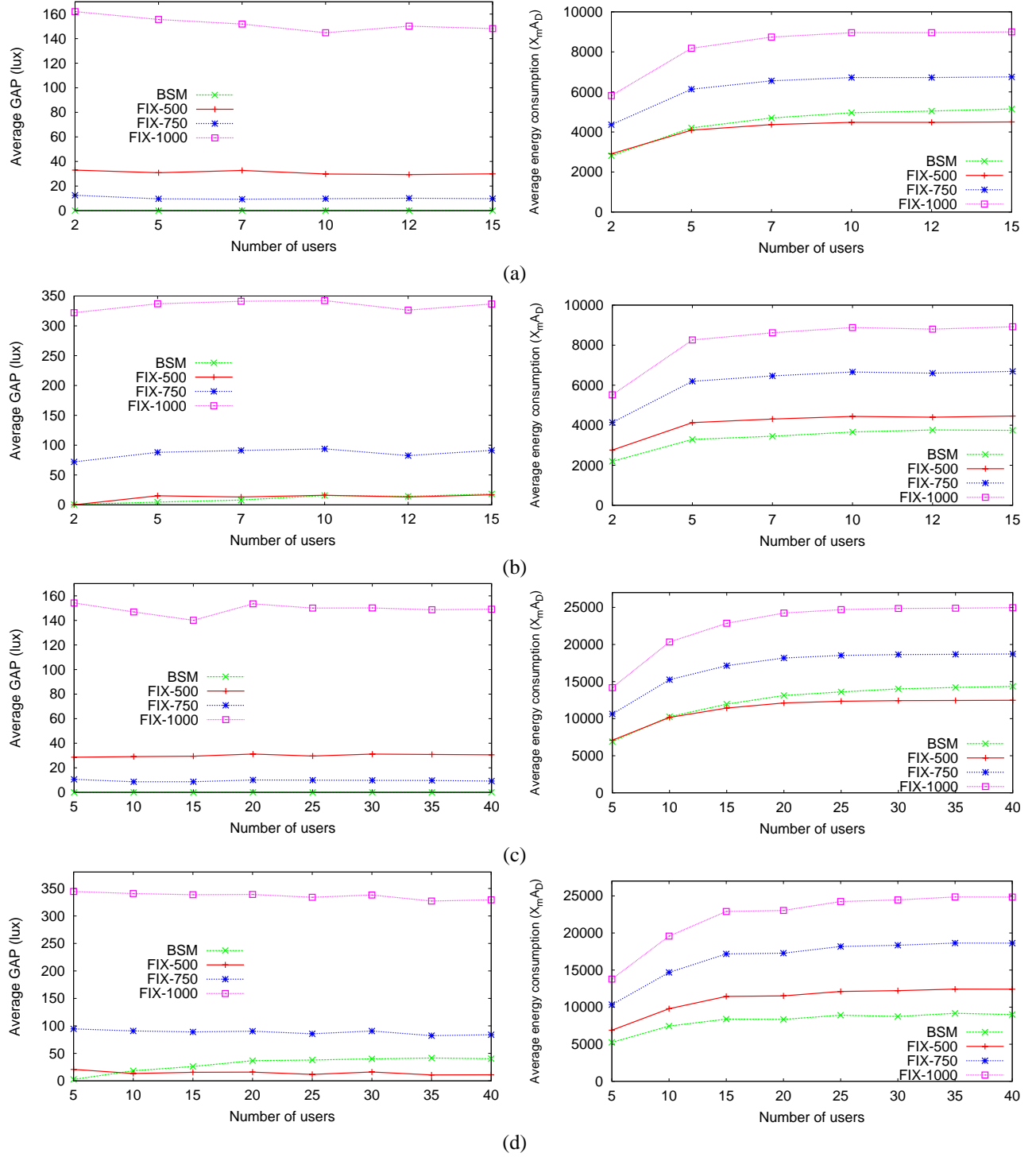


Fig. 13. Comparison of the proposed BSM and the FIX schemes when (a) the network scenario is $S1$ and the user-activity is $AR1$, (b) the network scenario is $S1$ and the user-activity is $AR2$, (c) the network scenario is $S2$ and the user-activity is $AR1$, and (d) the network scenario is $S2$ and the user-activity is $AR2$.

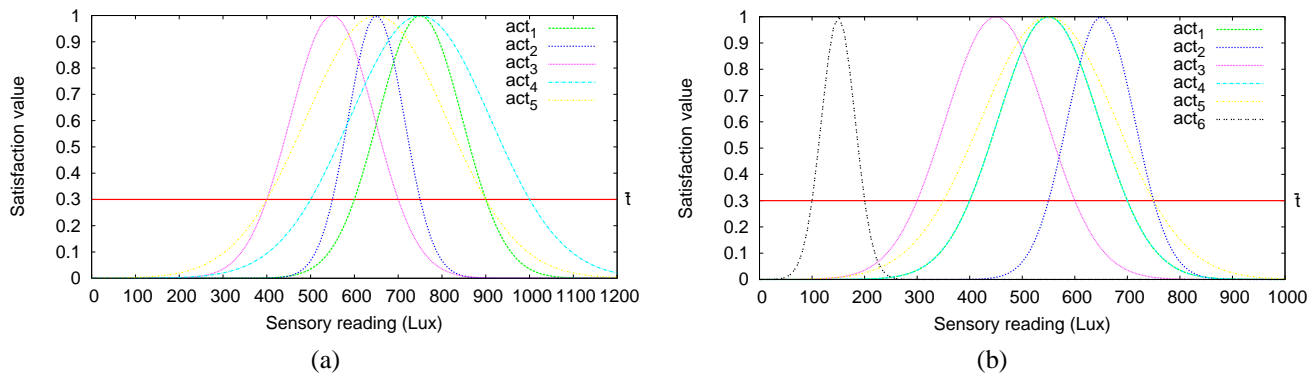


Fig. 14. Activity-requirement pools: (a) AR3 and (b) AR4.

- [16] H. Park, M. B. Srivastava, and J. Burke. Design and implementation of a wireless sensor network for intelligent light control. In *Proc. of ACM/IEEE Int'l Conference on Information Processing in Sensor Networks (IPSN)*, 2007.
- [17] N. Patwari, I. Alfred O. Hero, M. Perkins, N. S. Correal, and R. J. ODea. Relative location estimation in wireless sensor networks. *IEEE Trans. Signal Processing*, 51(8):2137–2148, 2003.
- [18] J. Sankaran. A note on resolving infeasibility in linear programs by constraint relaxation. *Operations Research Letters*, 13(1):19–20, 1993.
- [19] V. Singhvi, A. Krause, C. Guestrin, J. H. Garrett, and H. S. Matthews. Intelligent light control using sensor networks. In *Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, 2005.
- [20] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai. Wireless sensor networks for emergency navigation. *IEEE Computer*, 39(7):55–62, 2006.
- [21] Y.-J. Wen, J. Granderson, and A. M. Agogino. Towards embedded wireless-networked intelligent daylighting systems for commercial buildings. In *Proc. of IEEE Int'l Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2006.
- [22] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. of IEEE INFOCOM*, 2002.



Yen-Ann Chen received his B.S. and M.S. degrees from the National Dong Hwa University and National Chiao Tung University, in 2005 and 2007, respectively. His research interests include wireless communication and sensor networks.



Yu-Hsuan Lin received his B.S. degree from the National Chiao Tung University, in 2007. He is now pursuing the M.S. degree in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests include wireless communication and sensor networks.



Meng-Shiuan Pan received the B.S. and M.S. degrees from the National Chung Cheng University and National Tsing Hua University, Taiwan, in 2001 and 2003, respectively. Now, he is a graduate student pursuing the PhD degree in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests include mobile computing and wireless communication.



Lun-Wu Yeh received his B.S. and M.S. degrees in Computer and Information Science from the National Chiao-Tung University, in 2003 and 2005, respectively. He is currently pursuing PhD in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests include smart living space and wireless sensor network.



Yu-Chee Tseng received his B.S. and M.S. degrees in Computer Science from the National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He obtained his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. He was an Associate Professor at the Chung-Hua University (1994 1996) and at the National Central University (1996 1999), and a Professor at the National Central University (1999 2000). Currently, he is Chairman of the Department of Computer Science, and Associate

Dean of the College of Computer Science, National Chiao-Tung University, Taiwan.

Dr. Tseng received the Outstanding Research Award, by National Science Council, ROC, in both 2001-2002 and 2003-2005, the Best Paper Award, by Intl Conf. on Parallel Processing, in 2003, the Elite I. T. Award in 2004, and the Distinguished Alumnus Award, by the Ohio State University, in 2005. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Dr. Tseng is a member of ACM and a Senior Member of IEEE.

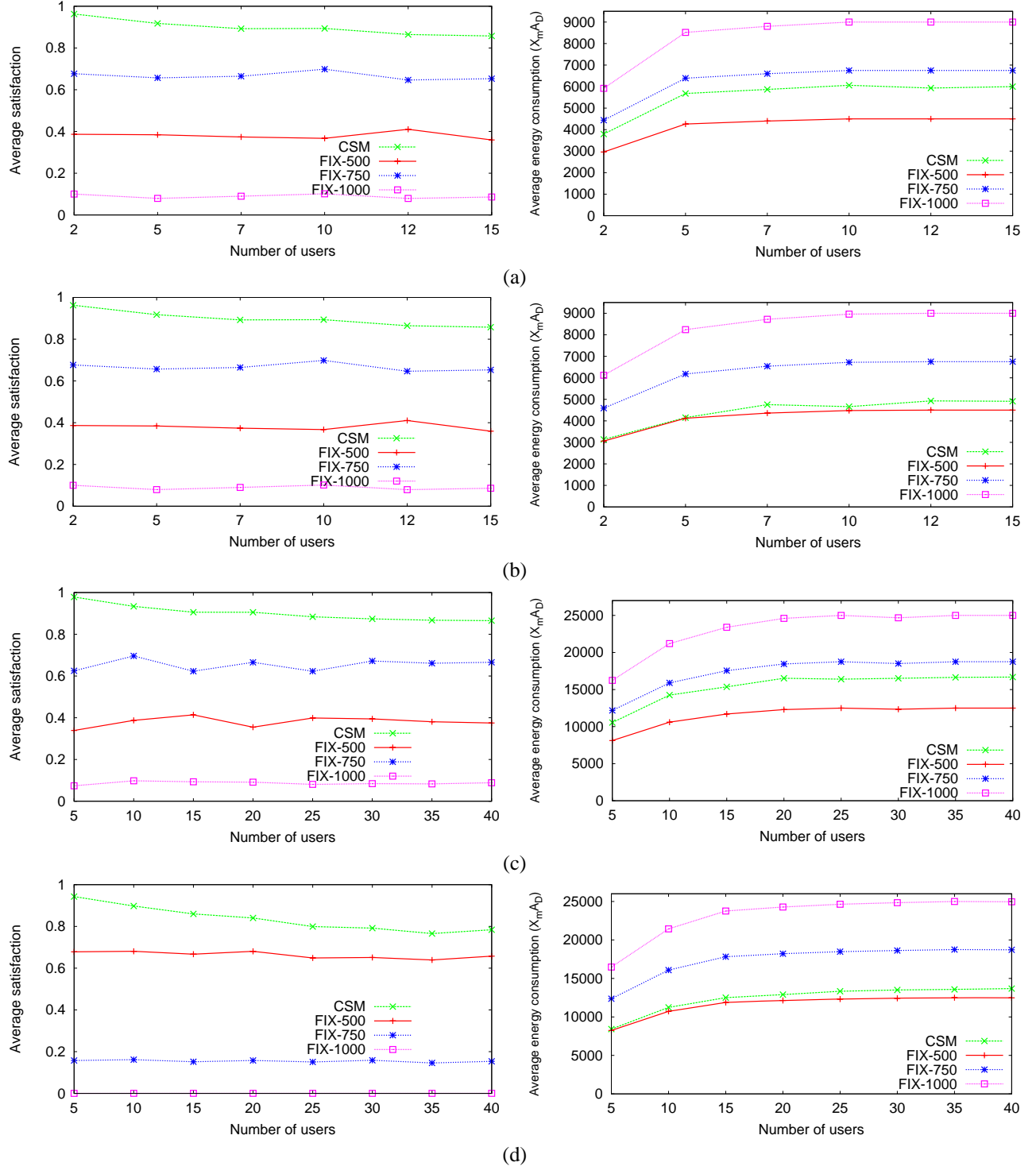


Fig. 15. Comparison of the proposed CSM and the FIX schemes when (a) the network scenario is S_1 and the user-activity is AR_3 , (b) the network scenario is S_1 and the user-activity is AR_4 , (c) the network scenario is S_2 and the user-activity is AR_3 , and (d) the network scenario is S_2 and the user-activity is AR_4 .